

⑬ 日本国特許庁 (JP) ⑭ 特許出願公開
⑯ 公開特許公報 (A) 昭59—165143

⑮ Int. Cl.³
G 06 F 9/34

識別記号

庁内整理番号
7218—5B

⑰ 公開 昭和59年(1984)9月18日

発明の数 1
審査請求 未請求

(全 3 頁)

⑱ データ処理装置

⑲ 特 願 昭58—39231

⑳ 出 願 昭58(1983)3月11日

㉑ 発 明 者 矢田 潔
秦野市堀山下1番地株式会社日
立製作所神奈川工場内

㉒ 発 明 者 沢田栄夫

秦野市堀山下1番地株式会社日
立製作所神奈川工場内

㉓ 出 願 人 株式会社日立製作所
東京都千代田区丸の内1丁目5
番1号

㉔ 代 理 人 弁理士 高橋明夫 外1名

明 細 書

1. 発明の名称

データ処理装置

2. 特許請求の範囲

(1) 命令語の一部がオペランドとなる即値命令の処理にあたり、上記命令語が指定する記憶装置のアドレスからオペランドを読出して保持しておくと共に、該オペランドと上記命令語の即値オペランドとを演算し、この演算結果と上記保持しておいたオペランドとを併合して上記記憶装置にストアすることを特徴とするデータ処理装置。

3. 発明の詳細な説明

(発明の利用分野)

本発明は命令語の一部がオペランドとなる即値命令を処理するデータ処理装置に関する。

(従来技術)

即値命令は、第1図にその命令形式を示すように、命令語のBフィールドで示すベースレジスタの内容とDフィールドのディスプレースメントを加算して得た記憶装置(MS)のアドレスの1バ

イトのデータ(オペランド)と、Iフィールドで示す1バイトの即値データとに対し、Fフィールドで示す演算を行い、その結果を上記BとDフィールドで示すMSの同じアドレスにストアするものである。この場合、MSのデータ幅は例えば8バイトであり、これに対して即値命令による演算は例えば1バイトでMSのデータ幅より小さいため、1バイトの部分書き込みを行なう。第2図にその処理フローを示す。これによれば、即値命令処理を行なうためには、MSからのオペランドフェッチ、演算、結果のMSへのストアサイクルと、計2度のMSサイクルと演算サイクルが必要であり、そのうちの結果のMSへのストアサイクルは1バイトの部分書き込みである。

ところで、部分書き込みはMSのフェッチサイクルとストアサイクルを必要とし、ストアするアドレスを含む8バイトを読出し、これにストアすべき1バイトを併合してストアする。このため結局、3度のMSサイクルと1度の演算サイクルを必要とするため、比較的使用頻度の高い即値命令とし

てはその処理速度が遅いという欠点があつた。

〔発明の目的〕

本発明の目的は、データ処理装置において即値命令の処理時間の短縮をはかることにある。

〔発明の概要〕

本発明は、命令語が指定する記憶装置のアドレスからオペランドを読出して保持しておくと共に、該オペランドと命令語の即値オペランドとを演算し、その結果を上記保持しておいたオペランドと併合して記憶装置にストアすることを特徴とする。

〔発明の実施例〕

第3図は本発明の一実施例であり、特にMS制御部と演算器を示す。即値命令の実行は、命令の準備ルーチンで、Rレジスタ9に命令語の1フィールドの即値オペランドが、また、BおよびDフィールドにより求められたオペランドアドレスはMS1のアドレスレジスタSAR2に設定される。SAR2のアドレスにより、MS1から該アドレスのデータを含む8バイトのデータを読出し、レジスタRDR3に読出す。この8バイトのデータ

は前述のストアまで保持されている。この8バイトのデータのうち、オペランドアドレスで指定した該当する1バイトのデータをスイッチ7により選択する。この1バイトのデータは演算器8に送られ、演算器8はこの1バイトのデータと、Rレジスタ9にセットされている即値オペランドとの演算を行ない、結果を書込みレジスタWDR4にセットする。この時、同時にオペランドアドレスに基づいて作成されたストアバイト位置を示す8バイトの書込みフラグを、書込みフラグレジスタMARK5にセットする。このMARK5で示すWDR4の1バイトとRDR3に先に読出されて保持されていた8バイトのうちのWDR4の選択された1バイト以外の7バイトがスイッチ6で併合され、8バイトの書込みデータとなつて先と同じMS1のアドレスにストアされる。

以上の一連の動作において、オペランドのフェッチから演算結果のストアまでMSを専有している。勿論、RDR3に読出したデータを別のレジスタで保持しておき、ストア時、WDR4の1バ

イトと別のレジスタに保持されている8バイトと併合してストアデータとすることにより、MSを専有しない方法をとることもできる。

一般にMSのアクセスは、処理装置CPUからのアクセス、チャネルからのアクセスを均等に処理するため、1つのフェッチ、ストアを単位として処理しており、本発明の様に、フェッチとストアをCPU内の演算サイクルを途中に介入させながら連続させるためには、即値命令によるオペランドフェッチ、即値命令によるストアをMSサイクル制御部に認識させる必要がある。即値命令によるオペランドフェッチでは、フェッチしたオペランドのRDR3での保持、または新たなMSサイクル起動（CPU以外の装置からの）の抑止を行ない、即値命令によるストアでは、通常の部分書込みはMARK5が全て“1”でない時であり、この時は、MSのフェッチ、ストアサイクルを連続させるが、即値命令のストアではMARK5が全て“1”でなくともMSストアサイクルのみ実行させる。またこの時のストアデータは前述の様に、

WDR4とRDR3をMARK5に従つて作成する。

以上をマイクロプログラム上で記述したのが第4図である。Aのマイクロ命令がオペランドフェッチを指示する。Bのマイクロ命令では、記憶保護等のMSアクセス例外のテストを行ない、Cではフェッチされたデータの演算器への転送と即値オペランドとの演算、演算結果のWDR4へのセットを行なう。Cのマイクロ命令は、MSからのフェッチデータが完了するまでその実行を待たす場合がある。DではMARKの作成とWDRとRDRによるストアデータのストアを指示する。

本実施例によれば、従来の即値命令処理の3MSサイクルと1演算サイクルを、2MSサイクルと1演算サイクルに短縮することが可能となる。

〔発明の効果〕

以上説明したように、本発明によれば、即値命令の処理にあつて、MSからの読出しデータを保持しておき、この保持しておいたデータに演算結果を併合してストアデータとすることにより、

部分書き込みのためのフェッチサイクルを不要とするので、処理時間の短縮をはかることができる。

4. 図面の簡単な説明

第1図は即値命令の形式を示す図、第2図は即値命令の処理フロー示す図、第3図は本発明の一実施例を示すブロック図、第4図は即値命令の処理マイクロプログラムフローを示す図である。

1…記憶装置MS、2…MSアドレスレジスタ、3…読出しレジスタ、4…書き込みレジスタ、5…書き込みフラグレジスタ、6…書き込みデータスイッチ、7…読出しデータスイッチ、8…演算器、9…即値オペランド保持レジスタ、10…ワークレジスタ。

代理人 弁理士 高橋 明 夫

図 1

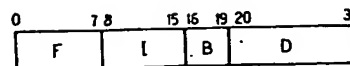


図 2

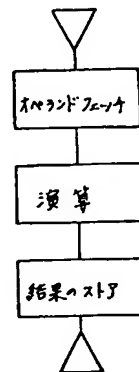


図 4

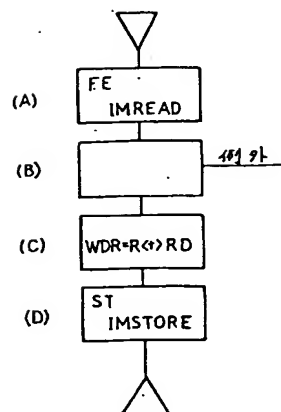
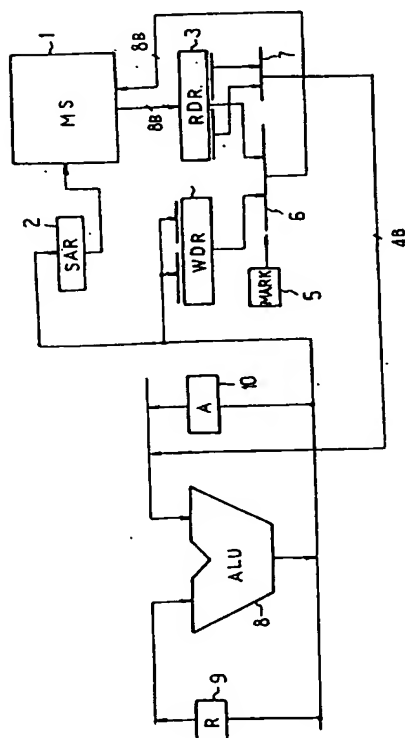


図 3



JAPANESE LAID-OPEN PATENT APPLICATION

S59-165143 (1984)

(19) Japan Patent Office (JP)

(11) Publication No. S59-165143

(12) Laid-Open Patent Application (A)

(43) Publication Date September 18, 1984

(51) Int. Cl.³

Identification Code

In-House Reference. No.

G 06 F 9/34

7218-5B

No examination request

Number of claims 1 (totally 3 pages)

(54) Title of the Invention

DATA PROCESSOR

(21) Application No.

PA S58-39231

(22) Date of Filing

March 11, 1983 (Showa 58)

(72) Inventor

Kiyoshi YATA

Hitachi Co., Ltd.

Kanagawa Plant

1, Horiyamashita

Hadano-shi

(72) Inventor

Eifu SAWADA

Hitachi Co., Ltd.

Kanagawa Plant

1, Horiyamashita

Hadano-shi

(71) Applicant

Hitachi Co., Ltd.

1-5-1, Marunouchi, Chiyoda-ku
Tokyo

(74) Agent

Akio TAKAHASHI, Attorney
(with another person)

Specification

I. Title of the Invention

Data Processor

II. Claims

(1) A data processor, characterized by the fact that, in processing the immediate instruction which becomes an operand, a portion of the instruction language is read from the address of a storage device indicated by the instruction language, and is held, in addition to which the operand and immediate operand of the instruction language are calculated, and the calculation results and the held operand are merged and stored in the storage device.

III. Detailed description of the invention

[Applied field of the invention]

The present invention relates to a data processor for processing immediate instructions in which a part of instruction language becomes an operand.

[Prior Art]

As the instruction format of the immediate instructions shown in Fig. 1, the immediate instructions perform the calculations shown in an F field relative to the immediate data of 1 byte shown by the data

(operand) of 1 byte of the address of a storage device (MS), and in the I field, obtained by adding the contents of a base register shown in the B field of the instruction language and a displacement of the D field, as shown in the instruction format in Figure 1. The results are stored in the same address of an MS shown by the B and D fields. In this instance, the data width of MS is, for example, 8 byteytes, while the calculation based on the immediate data is, for example, 1 byte and smaller than the data width of MS. Therefore a 1 byte partial-write is performed, the processing width of which is shown in Fig. 2. By this flow, the cycles of an operand fetch from MS, and the calculation and storage of the results into MS, 2 MS cycles in total and 1 calculation cycle, are necessary for performing immediate instruction processing, and the cycle of storing the results into MS is a 1 byte partial-write.

The partial-write needs the fetch cycle and the storage cycle of MS, 8 byteyte data including the storage address is read and 1 byteto be stored is merged in and stored. Therefore, there has been the draw-back that the processing speed was slow for immediate instructions having a higher use frequency because 3 MS cycles and 1 calculations cycle are needed after all.

[Purpose of the Invention]

The purpose of the present invention consists in shortening the processing time of the immediate instructions in a data processor.

[Summary of the Invention]

The present invention is characterized by reading an operand from the address of a storage device for assigning instruction language and holding it, computing the storage buffer operand and immediate operand of the storage buffer instruction language and merging the results with the above holding operand and storing them in the above memory device.

[Embodiment of the Invention]

Fig. 3 is an Embodiment of the present invention, and particularly shows an MS control unit and a calculation-logic unit. In the execution of immediate instructions, an immediate operand of I field of instruction language and an operand address obtained by B and D fields in an address SAR 2 of MS 1 in an R register 9 are set up by a preparatory routine of instructions. Eight byte data including the data of a storage buffer address are read from MS 1 according to the address of MS 1 and read the data into a register RDR 3. This 8 byte data is held until storage described later. Corresponding 1 byte data assigned by the operand address in the 8 byte data is selected with a switch 7. The 1 byte data is sent to a calculations-logic unit 8, the calculations-logic unit 8 performs calculations of the 1 byte data and an immediate operand which has been set up in the R register 9, and the results are set up in a write register WDR 4. At this time, an 8 byte write flag showing a store byte position simultaneously prepared based on an operand address is set up in a write flag register MARK 5. One byte of WDR 4 represented by the MARK 5 and 7 bytes of other than 1 byte of WDR 4 selected in the 8 bytes formerly read and

held in RDR 3 are merged with a switch 6, become an 8 byte write data and are stored in an address of MS 1 the same as before.

In a series of the above actions, MS is exclusive from the fetch of an operand to the storage of the calculation results. The data read in RDR 3 is held in another register, and a method of non-exclusive MS can also be adopted by merging the 1 byte data of WDR 4 and the 8 bytes held in another register as storage data.

Generally, access from a processing unit CPU and access from a channel in accessing MS are equally processed. Therefore the accesses are processed by adopting one fetch and one storage as a unit, and like the present invention, an operand fetch by an immediate instruction and a storage by an immediate instruction must be recognized in an MS cycle control unit in order to continue the fetch and the storage while interposing a calculations cycle in CPU on the way. In an operand fetch by an immediate instruction, the holding of a fetched operand in RDR 3 or the suppression of the start of a new MS cycle (from a unit other than a CPU) is performed; in the storage by an immediate instruction, the common partial-write is at a time in which not all MARK 5 are "1". At this time, the fetch and the storage cycles of MS are continued, but not all MARK 5 are "1" and only the MS storage cycle is executed in the storage of an immediate instruction. As the storage data at this time, WDR 4 and RDR 3 are prepared in accordance with MARK 5 as described above.

Fig. 4 describes the above in a micro-program. A micro-instruction of A indicates an operand fetch. A test on MS access exception such as storage protection, etc. is performed by a micro-instruction of B, and a transfer of fetched data to the calculations-logic unit and

calculations of an immediate operand are performed by a micro-instruction of C. The micro-instruction of C sometimes waits for its execution until the data fetch from MS is completed. The preparation of MARK and the storage of storage data by WDR and RDR are indicated by D.

The embodiment enables shortening 3 MS cycles and 1 calculation cycle of conventional immediate instruction processing to 2 MS cycles and 1 calculation cycle.

[Efficacy of the invention]

As described above, the present invention enables shortening the processing time in the processing of immediate instructions because the fetch cycle for a partial-write is not needed by holding the read data from MS and merging the calculations results with the held data as storage data.

IV. Brief description of the drawings

Fig. 1 is a diagram showing a format of immediate instruction,
Fig. 2 is a diagram showing a processing flow of immediate instruction,
Fig. 3 is a block diagram showing one Embodiment of the present invention, and
Fig. 4 is a diagram showing a processing micro-program flow of an immediate instruction.

- | | |
|----|------------------------------------|
| 1 | memory system MS |
| 2 | MS address register |
| 3 | read register |
| 4 | write register |
| 5 | write flag register |
| 6 | write data switch |
| 7 | read data switch |
| 8 | calculations-logic unit |
| 9 | immediate operand holding register |
| 10 | work register |

Fig. 1

Fig. 2

Operand fetch
|
Calculations
|
Storage of data

Fig. 3

Fig. 4

(right) exception

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.